UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,383	12/31/2001	Sushma Shrikant Trivedi	4860.P2692	7758

7590 05/31/2005

James C. Scheller
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 05/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/038,383	TRIVEDI ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 March 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-87* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-87* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-87 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 3/11/2005.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-87 are rejected under 35 U.S.C. 102(e) as being anticipated by Dowling, U.S. Patent No. 6,363,475 (as applied in the previous Office Action).

5. Referring to claims 1, 28, and 55, Dowling has taught a method for dispatching instructions executed by at least one functional unit of a data processor, each one of the instructions having a corresponding priority number, in a computer system having at least one host processor and host memory, the method comprising:

a) in response to a next instruction, examining a current instruction group to determine if the current instruction group is completed, each instruction in the current instruction group

associated with a priority number corresponding to a type of functional unit executing the respective instruction, and wherein the current instruction group is formed based on a priority number of each instruction in the current instruction group and availability of the corresponding functional unit. See column 14, lines 3-23, and note that after group of instructions are fetched, they are broken into execute packets. Hence, each instruction is inherently analyzed to determine which execute packet the instruction belongs in. Execute packets are executed sequentially, i.e., the packet as a whole, and consequently, each of the instructions within the packet, have a priority. The instructions of the execute packet to be executed first have the highest priority. The instructions of the execute packet to be executed second have the second highest priority, and so on. In addition, the grouping is based on availability of functional units. See column 1, lines 49-55, and note that instructions are grouped so that the highest possible amount of instructions may execute concurrently on parallel functional units. So, overall, instructions are grouped based on execution unit availability (an instruction cannot be grouped and issued if a functional unit to execute it is not available) and based on priority (instructions that appear in the program earlier are executed earlier).

b) adding the next instruction to the current instruction group if the current instruction group is not completed. See column 14, lines 3-23, and note the example where there are four groups of 2 instructions each. As instructions are fetched, they are put into execute packets. So, when a first instruction is encountered, it will be put in a first execute packet. When a second instruction is encountered, if it is able to be grouped with the first instruction, then the second instruction will be added to the first instruction's execute packet since there is still an opening in the packet.

c) dispatching the current instruction group if the current instruction group is completed, wherein the current instruction group is completed when one or more predetermined conditions are satisfied. See column 14, lines 3-23, and note that execute packets are completed groups of instructions which are to execute concurrently. Consequently, an execute packet will be dispatched for execution. The predetermined condition would be grouping all eligible instructions that may be grouped. Once this happens, the execute packet is complete and dispatching may occur.

6. Referring to claims 2, 29, and 56, Dowling has taught the method of claim 1, wherein if the current instruction group is completed, the method further comprises starting a new instruction group and adding the next instruction to the new instruction group. Again, see column 14, lines 3-23, and note that if in one particular example, instruction groups comprise two instructions, then when two instructions have already been grouped, a third instruction will be put in a new, next packet.

7. Referring to claims 3, 30, and 57 Dowling has taught the method of claim 1, further composing:

a) examining the next instruction to determine if the corresponding priority number of the next instruction is equal to or lower than the corresponding priority number of a current instruction of the current instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

b) adding the next instruction to the current instruction group if the corresponding priority number of the next instruction is higher than the corresponding priority number of the current instruction of the current instruction group (Dowling abstract, figures 4-5, column 3 line 54-

column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). Again, if an instruction has higher priority than a current instruction (it precedes the current instruction in program order), then it may be put in the same group, if there's room.

c) dispatching without adding the next instruction the current instruction group if the corresponding priority number of the next instruction is equal to or lower than the corresponding priority number of the current instruction of the current instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). If a group is full, and an instruction that should execute after the instructions of the current group is encountered, then that instruction is put in a new group since it is of lower priority (it is to execute later). In addition, with the current group being full, it may be dispatched for execution.

8. Referring to claims 4, 31, and 58, Dowling has taught the method of claim 3, wherein if the corresponding priority number of the next instruction is higher than the corresponding priority number of the current instruction of the current instruction group, the method further comprises:

examining the next instruction to determine if the next instruction is required to be in a new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

wherein if the next instruction is required to be in a new instruction group:

adding a no-operation (NOOP) instruction to the current instruction group to complete the current instruction group. See column 7, lines 24-39, and column 11, lines 45-49. Note that NOPS are inserted into the packets for alignment purposes, when necessary.

dispatching the current instruction group without adding the next instruction. (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). If a group is full, and an instruction that should execute after the instructions of the current group is encountered, then that instruction is put in a new group since it is of lower priority (it is to execute later). In addition, with the current group being full, it may be dispatched for execution.

starting a new instruction group; and

adding the next instruction to the new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

9. Referring to claims 5, 32, and 59, Dowling has taught the method of claim 3, wherein if the corresponding priority number of the next instruction is higher than the corresponding priority number of the current instruction of the current instruction group, the method further comprises:

examining the current instruction group to determine if the current instruction group contains a predetermined number of instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

wherein if the current instruction group contains the predetermined number of instructions:

dispatching the current instruction group as a completed group. Note that when an execute packet is dispatched, it is completed.

starting a new instruction group; and

adding the next instruction to the new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

10. Referring to claims 6, 33, and 60, Dowling has taught the method of claim 1, further comprising:

examining the current instruction group to determine if the current instruction group contains a predetermined number of instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23); and

dispatching the current instruction group if the current instruction group contains the predetermined number of instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

11. Referring to claims 7, 34, and 61, Dowling has taught the method of claim 1, wherein all instructions in the current instruction group are dispatched in the same clock cycle (Dowling column 3 lines 61-63).

12. Referring to claims 8, 35, and 62 Dowling has taught the method of claim 1, further comprising:

a) examining the next instruction to determine latency required by the next instruction, wherein the latency is determined based on a type of a functional unit executing the next instruction. See column 12, lines 21-40, and note that when a conditional branch instruction is to be executed by

a branch execution unit, a latency is determined (the latency being associated with the amount of time it takes to execute both branch paths).

b) calculating delay cycles based on the latency. See column 12, lines 21-40, and note that one path may finish execution sooner than the other. The difference in execution cycles is determined so that NOPs may be inserted into the already finished path.

c) suspending the dispatching for a period of time corresponding to the delay cycles. No real instruction groups would be further dispatched until the other path has finished executing. Instead, NOPs will be sent to the functional units, thereby allowing the sub-processor executing along the finished path to synchronize with the second path (Dowling column 12 lines 21-40).

13. Referring to claims 9, 36, and 63 Dowling has taught the method of claim 8, further comprising inserting an additional delay cycle during the suspension (Dowling column 12 lines 21-40).

14. Referring to claims 10, 37, and 64 Dowling has taught the method of claim 1, further comprising:

examining the next instruction to determine if the next instruction contains an illegal operation code (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23); and

issuing an error message through an interrupt mechanism, if the next instruction contains an illegal operation code (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

15. Referring to claims 11, 38, and 65 Dowling has taught the method of claim 1, wherein if the next instruction is a non-branch instruction, the method further comprises:

examining the next instruction to determine if source resources required by the next instruction are in-use (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26); and

stalling instruction dispatching if the source resources required by the next instruction are in-use (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

16. Referring to claims 12, 39, and 66 Dowling has taught the method of claim 11, wherein the source resources are defined by source operand registers required by the next instruction (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

17. Referring to claims 13, 40, and 67 Dowling has taught the method of claim 1, wherein if the next instruction is a non-branch instruction, the method further comprises:

examining the next instruction to determine if destination resources required by the next instruction are in-use (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26); and

stalling instruction dispatching if the destination resources required by the next instruction are in-use (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

18. Referring to claims 14, 41, and 68 Dowling has taught the method of claim 13, wherein the destination resources are defined by target destination registers required by the next instruction (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

19. Referring to claims 15, 42, and 69 Dowling has taught the method of claim 1, wherein if the next instruction is a branch instruction, the method further comprises:

examining resources required by the branch instruction to determine if the resources are used **or altered by a non-branch instruction**.

wherein if the resources are used **or altered by a non-branch instruction**, suspending the dispatching the next instruction until the resources are available. It should be noted that applicant is claiming how a VLIW system works. That is, when a conditional branch is encountered, the system will either use resources for executing the target path or the fall-through path. Whatever path is taken, resources will still be used for execution. If the resources being used are required to execute a next instruction, then the next instruction will have to wait until the resources are free (dispatch is suspended). Applicant should note his/her use of the "or" clause. In essence, the examiner is ignoring the phrase "or altered by a non-branch instruction" for purposes of examination, as the prior art inherently teaches everything before the "or" clause, and that is all that is required to anticipate the claim.

20. Referring to claims 16, 43, and 70 Dowling has taught the method of claim 15, further comprising inserting an additional delay cycle during the suspension (Dowling abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

21. Referring to claims 17, 44, and 71 Dowling has taught the method of claim 5, wherein the predetermined number of instructions comprises four instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

22. Referring to claims 18, 45, and 72 Dowling has taught the method of claim 6, wherein the predetermined number of instructions comprises four instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

23. Referring to claims 19, 46, and 73 Dowling has taught the method of claim 3, further comprising accessing a database to determine the corresponding priority number of the next instruction (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

24. Referring to claims 20, 47, and 74 Dowling has taught the method of claim 8, further comprising accessing a database to determine the latency required by the next instruction (Dowling column 12 lines 21-40).

25. Referring to claims 21, 48, and 75 Dowling has taught the method of claim 1, wherein the data processor is integrated in a system core logic chip that functions as a bridge between the host processor and the host memory, and other components of the computer system, the system core logic chip having a host interface coupled to the host processor and a memory interface

coupled to the host memory (Dowling abstract, figures 2, 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

26. Referring to claims 22, 49, and 76 Dowling has taught the method of claim 1, wherein the data processor may be a stand-alone processor, or the data processor may be a co-processor to the host processor (Dowling abstract, figures 2, 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

27. Referring to claims 23, 50, and 77 Dowling has taught the method of claim 1, wherein the at least one functional unit comprises multiple functional units of a kind (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

28. Referring to claims 24, 51, and 78 Dowling has taught the method of claim 23, further comprising:

examining the next instruction to determine if there is a corresponding functional unit that executes the next instruction available (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

adding the next instruction to the current instruction group if the corresponding functional unit is available (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). The purpose of VLIW is to execute as many

instructions as possible in parallel. Therefore, if a unit is available and the next instruction uses that unit, then it will be added to the group.

dispatching the current instruction group without adding the next instruction if the corresponding functional unit is not available (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). If a group is full, then no more functional units are available. Consequently, with the current group being full, it may be dispatched for execution, while any subsequent instruction would be put in new groups.

29. Referring to claims 25, 52, and 79 Dowling has taught the method of claim 24, wherein if the corresponding functional unit that executes the next instruction is available, the method further comprises:

examining the next instruction to determine if the next instruction is required to be in a new instruction group, wherein the next instruction is required to be in the new instruction group if a resource used by the next instruction is to be used by at least one instruction of the current instruction group or the next instruction is in a category different than at least one instruction of the current instruction group. (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). It should be realized that applicant is claiming how a VLIW processor works. That is, if there is a single multiply unit, which takes one multiply instruction per cycle, then if a group already contains a multiply instruction, another multiply instruction must be put in a new group because the resource it requires is already allocated to the current group.

wherein if the next instruction is required to be in a new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

adding a no-operation (NOOP) instruction to the current instruction group to complete the current instruction group. The examiner asserts that this limitation explains how a VLIW operates. If a particular slot of a VLIW cannot be filled, then a NOP is put into it, as it specifies that no operation is to be performed.

dispatching the current instruction group without adding the next instruction. (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23). If a group is full, then no more functional units are available. Consequently, with the current group being full, it may be dispatched for execution, while any subsequent instruction would be put in new groups.

starting a new instruction group; and

adding the next instruction to the new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

30. Referring to claims 26, 53, and 80, Dowling has taught the method of claim 24, wherein if the corresponding functional unit that executes the next instruction is available, the method further comprises:

examining the current instruction group to determine if the current instruction group contains a predetermined number of instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

wherein if the current instruction group contains the predetermined number of instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

dispatching the current instruction group without adding the next instruction (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

starting a new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23); and

adding the next instruction to the new instruction group (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

31. Referring to claims 27, 54, and 81, Dowling has taught a method of claim 26, wherein the predetermined number of instructions comprises four instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

32. Referring to claim 82, Dowling has taught an apparatus for dispatching instructions executed by at least one functional unit of a data processor, the apparatus comprising:

an instruction cache memory for receiving instructions from an input and output (1/0) interface (Dowling abstract, figures 2 and 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23);

an instruction decoder coupled to construct an instruction group based on the priorities of the instructions (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23); and

a dispatch controller coupled to dispatch the instruction group to an appropriate functional unit (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

33. Referring to claim 83, Dowling has taught the apparatus of claim 82, further comprising:

at least one instruction registers coupled to store the instructions being grouped; and

at least one instruction buffers coupled to store instructions when the instruction fetching is stalled (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

34. Referring to claim 84, Dowling has taught the apparatus of claim 82, further comprising a branch decoder coupled to detect a branch condition and to generate the address for the next instruction being fetched (Dowling figure 7, abstract, column 3 lines 13-30, column 7 lines 3-23).

35. Referring to claim 85, Dowling has taught the apparatus of claim 84, further comprising a program counter coupled to receive commands from the branch decoder to fetch the next instruction at the address (Dowling figure 7, abstract, column 3 lines 13-30, column 7 lines 3-23).

36. Referring to claim 86, Dowling has taught the apparatus of claim 83, wherein the instruction decoder retrieves the instructions from the at least one instruction registers or from

the at least one instruction buffers after the instruction stalling cycles (Dowling column 12 lines 21-40, abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

37. Referring to claim 87, Dowling has taught the apparatus of claim 82, wherein the instruction decoder stalls the instruction fetching based on the latency of the instruction being executed (Dowling column 12 lines 21-40, abstract column 8 lines 40-58, column 10 line 63-column 11 line 26).

38. Claims 1-2, 28-29, and 55-56 are rejected under 35 U.S.C. 102(b) as being anticipated by Schepers, U.S. Patent No. 5,712,996.

39. Referring to claims 1, 28, and 55, Schepers has taught a method for dispatching instructions executed by at least one functional unit of a data processor, each one of the instructions having a corresponding priority number, in a computer system having at least one host processor and host memory, the method comprising:

a) in response to a next instruction, examining a current instruction group to determine if the current instruction group is completed, each instruction in the current instruction group associated with a priority number corresponding to a type of functional unit executing the respective instruction, and wherein the current instruction group is formed based on a priority number of each instruction in the current instruction group and availability of the corresponding functional unit. See the abstract, Tables 1 and 2 (in column 6, and the process of claim 1. Essentially, instructions are divided into groups based on priority, where priority is determined based on a number of factors, the factors being delay cycles, succeeding instructions, and distance value. See column 4, lines 33-51. In addition, it is inherent that a group must be

examined for completion because if it is completed (i.e., 3 instructions have been assigned to it), then a next instruction cannot be assigned to the group.

b) adding the next instruction to the current instruction group if the current instruction group is not completed. See Table 2 and column 6, line 48, to column 7, line 13, as an example of this. Basically, a first instruction (l.s) is assigned to a first group and a next instruction (lw) is subsequently assigned to the first group since the first group is not yet completed (i.e., there are not 3 instructions assigned to it yet).

c) dispatching the current instruction group if the current instruction group is completed, wherein the current instruction group is completed when one or more predetermined conditions are satisfied. See the abstract and Table 2 and note that when instruction groups are completed, i.e., they have 3 instructions assigned to them, then they are dispatched for execution. One predetermined condition would be whether or not the group has 3 instructions assigned to it.

40. Referring to claims 2, 29, and 56, Schepers has taught a method as described in claim 1. Schepers has further taught that if the current instruction group is completed, the method further comprises starting a new instruction group and adding the next instruction to the new instruction group. Again, if there is no more room to add an instruction to a group a new group is started and a next instruction is added.

Response to Arguments

41. Applicant's arguments filed on March 11, 2005, have been fully considered but they are not persuasive.

42. Applicant argues the novelty/rejection of claim 1 on page 25 of the remarks, in substance that:

> "Although Dowling discloses dispatching different group of instructions from multiple instruction streams, Dowling still fails to disclose the groups of instructions is dispatched only when the group is completed satisfying one or more predetermined conditions. In addition, although Dowling discloses assigning a priority to a steam of instructions, Dowling fails to disclose that each instruction is associated with a priority number corresponding to the type of functional unit for executing the respective instruction. See, for example, col. 3, line 31 to col. 4, line 7 of Dowling. Therefore, for the reasons discussed above, it is respectfully submitted that independent claim 1 is not anticipated by Dowling."

43. These arguments are not found persuasive for the following reasons:

a) Looking at column 14, lines 3-23, it can be seen that execute packets are formed from a fetch packet and that a first execute packet is dispatched in a first cycle, a second execute packet is dispatched in a second cycle, and so on. The examiner asserts that when the execute packets (groups) are completed, then they are dispatched. In the example given, with a fetch packet of 8 instructions, four groups of 2 instructions may be formed. Each group is complete when it has two instructions, and then it is dispatched.

b) In addition, with regards to priority, if a particular thread has priority, then it follows that its individual instructions also have priority over instructions of another thread having lower priority. And, priority exists among individual instructions as well. That is, instructions inherently issue at different times. The instructions issued first have higher priority over instructions that issue later (this is the reason they issue first).

44. Applicant argues the novelty/rejection of claim 3 on page 26 of the remarks, in substance that:

> "It is respectfully submitted that Dowling still fails to disclose the limitations set forth above [in claim 3]."

45. These arguments are not found persuasive for the following reasons:

a) As the examiner previously stated, priority inherently exists among instructions even if there is no explicit mention of priority. Instructions that are to execute sooner have higher priority than those executing later. Consequently, if an instruction has higher priority than an instruction in a current group, then the high priority instruction should be grouped in the current group. If however, the instruction has lower priority than the instruction in a current group, then it will go into a next group, if the current group is full.

46. With respect to applicant's arguments for claims 8 and 35 and 15 and 42, please see the above rejections of these claims as the examiner has clarified his position such that it responds to the arguments.

47. Applicant argues the novelty/rejection of claim 3 on page 26 of the remarks, in substance that:

> "There is no disclosure or suggestion within Dowling that the subprocessor shown in Fig.2 is implemented within a bridge between a host processor and a host memory, and the rest of the components of a data processing system.

48. These arguments are not found persuasive for the following reasons:

a) As disclosed in Dowling, component 200 of Fig.2 is a subprocessor. A subprocessor is a portion of the host processor which would inherently receive signals from the rest of the host processor (control/data/power signals). Therefore, it would have a host processor interface (i.e., it can communicate with the rest of the host processor). In addition, the subprocessor receives instructions from memory as shown in Fig.2, and consequently, has a memory interface. Since

the subprocessor receives data from the memory and from the rest of the host processor, it can be said that the subprocessor is between the host processor and memory (a bridge).

Conclusion

49. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

50. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Simar et al., EP Application Publication 0855648 A2, has taught fetching a packet of instructions having priority (from small address to large address) and grouping the instructions based on p-bit value within the fetch packet.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
May 24, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100